

CLAIM AMENDMENTS

This listing of claims will replace all prior versions, and listings, of claims in the application.

1 1. (Currently Amended) A data processing system for executing a program
2 having branch instructions therein, each branch instruction specifying a target
3 address in said program defining an instruction that is to be executed if that
4 branch instruction causes said program to branch, said data processing system
5 comprising:

6 a plurality of processing sections, each processing section comprising:

7 a local memory for storing instruction sequences from said
8 program that are to be executed by that processing section, said
9 instruction sequences comprising instructions of different lengths;

10 a function unit for executing instructions stored in said local
11 memory; and

12 a pointer containing a value defining the next instruction in said
13 local memory to be executed by said function unit, wherein each
14 processing section executes part of said program;

15 each function unit executes instructions according to machine
16 cycles, each function unit executing one instruction per machine cycle
17 ~~synchronously with said other function units~~; and

18 said pointers in each of said processing sections are reset to a
19 new value determined by said target address of one of said branch
20 instructions when a function unit branches in response to that branch
21 instruction.

1 2. (Currently Amended) The data processing system of claim 1 further
2 comprising a memory for determining said new value of said pointers, said
3 memory storing a mapping for each target address in said program specifying
4 one of said pointer values for each of said pointers corresponding to that target
5 address.

1 3. (Currently Amended) The data processing system of claim 1 wherein said
2 program is divided into super instructions, each super instruction comprising a
3 linear block of code that can only be entered at a starting address
4 corresponding to said block of code and each block of code having one or
5 more branch instructions, wherein said target address of at least one of said
6 branch instructions ~~having a target address corresponding to~~ corresponds to a
7 starting address of a super instruction in said program.

1 4. (Currently Amended) The data processing system of claim 1 wherein at
2 least one of said instruction sequences comprises at least one no op instruction
3 ~~one of said super instructions comprises one instruction for each processing~~
4 ~~section to be executed on each clock cycle.~~

1 5. (Original) The data processing system of claim 1 wherein said local
2 memory of one of said processing sections comprises a cache memory.

1 6. (New) The data processing system of claim 3 further comprising at least
2 one processing section that remains idle for the duration of a super instruction.

1 7. (New) The data processing system of claim 3 wherein a super instruction
2 comprises a tuple for each processing section that is not idle for the duration of
3 said super instruction and wherein each tuple identifies a function unit on
4 which execution occurs and a number of memory words needed to represent
5 corresponding operations to be executed on said function unit.

1 8. (New) The data processing system of claim 3 wherein a super instruction
2 indicates a total number of machine cycles for said super instruction.

1 9. (New) The data processing system of claim 3 wherein said local memories
2 are loaded at a start of said program.

1 10. (New) The data processing system of claim 3 wherein said local
2 memories are loaded at the time of a branch to a super instruction.

1 11. (New) The data processing system of claim 3 wherein a fall-through
2 super instruction is executed when a super instruction executes without
3 branching.

1 12. (New) A data processing system for executing a super instruction
2 according to machine cycles, said super instruction comprising a linear block
3 of code including instruction sequences to be executed by each of a plurality
4 of processing sections, one instruction for each machine cycle, the data
5 processing system comprising:
6 a plurality of processing sections, each processing section comprising:
7 a local memory for storing instruction sequences that are to be
8 executed by that processing section;
9 a function unit for executing instructions stored in said local
10 memory according to machine cycles, each function unit executing one
11 instruction per machine cycle; and
12 a pointer containing a value defining the next instruction in said
13 local memory to be executed by said function unit.

1 13. (New) The data processing system of claim 12 wherein said linear block
2 of code of said super instruction can only be entered at a starting address and
3 includes one or more branch instructions.

1 14. (New) The data processing system of claim 13 wherein said pointers in
2 each of said processing sections are reset to a new value determined by a
3 target address of one of said branch instructions when a function unit branches
4 in response to that branch instruction.

1 15. (New) The data processing system of claim 14 wherein said target
2 address of at least one of said branch instructions corresponds to a starting
3 address of a super instruction in said program.

1 16. (New) The data processing system of claim 14 further comprising a
2 memory for determining said new value of said pointers, said memory storing

3 a mapping for each target address specifying one of said pointer values for
4 each of said pointers corresponding to that target address.

1 17. (New) The data processing system of claim 12 wherein at least one of
2 said instruction sequences comprises at least one no op instruction.

1 18. (New) The data processing system of claim 12 said instruction sequences
2 comprise instructions of different lengths.

1 19. (New) The data processing system of claim 12 further comprising at least
2 one processing section that remains idle for the duration of the super
3 instruction.

1 20. (New) The data processing system of claim 12 wherein said super
2 instruction comprises a tuple for each processing section that is not idle for the
3 duration of said super instruction and wherein each tuple identifies a function
4 unit on which execution occurs and a number of memory words needed to
5 represent corresponding operations to be executed on said function unit.

1 21. (New) The data processing system of claim 12 wherein said super
2 instruction indicates a total number of machine cycles for said super
3 instruction.

1 22. (New) The data processing system of claim 12 wherein said local
2 memories are loaded at a start of a program that includes said super
3 instruction.

1 23. (New) The data processing system of claim 12 wherein said local
2 memories are loaded at the time of a branch to said super instruction.

1 24. (New) The data processing system of claim 12 wherein a fall-through
2 super instruction is executed when said super instruction executes without
3 branching.